Supplementary Information for:

What makes different people's representations alike: neural similarity-space solves the problem of across-subject fMRI decoding

Rajeev D. S. Raizada[1]* and Andrew C. Connolly[2]

[1]Department of Human Development, Cornell University, Ithaca NY 14853.

[2]Dept. of Psychological & Brain Sciences, Dartmouth College, Hanover NH 03755.

*To whom correspondence should be addressed: raizada@cornell.edu

Computer code, for easy replication and verification of our analyses

To run the analyses in the main text, download the accompanying code from the Supporting Information website. It can also be downloaded from
http://dl.dropbox.com/u/700503/RaizadaConnolly_SuppInfo_code_revised.zip

Then download the Haxby data, from http://pymvpa.org/datadb/haxby2001.html

The first set of analyses in the paper use that data without any additional preprocessing (the brain volumes in the online archive have already been motion-corrected).

To read in the fMRI data and write its similarity matrices to a Matlab-format file, run the Python script Step1_ReadHaxbyData_WriteMatlabSimMatrixSquareforms.py, which uses functions from PyMVPA (Hanke et al., 2009). Next, to perform the across-subject decoding, run the Matlab script Step2_AcrossSubjDecoding_SimMatrixPermutationMatching.m

Feature-selection, and similarity analyses of the feature-selected voxels

The code for the second set of analyses in the paper can be found in the subdirectory
Spatially_normalised_analyses.

In order to compare the locations of the selected voxels across different subjects, the brain volumes were all spatially normalised to the standard MNI152 template at 3x3x3mm resolution using SPM8, before feature-selection or similarity-analysis was carried out. A batch script to run those standard preprocessing steps is Step1_split_and_spatially_normalise_haxby_data.m.

The normalised images are then detrended and written into Matlab-format .mat files by the Python script Step2_haxby_normed_data_whole_brain_to_mat.py, which uses the whole-brain mask image, also provided, wholebrain_mask_haxby_space.nii.gz

The feature-selection is performed by the Matlab script
Step3_select_wholebrain_haxby_voxels_via_Ts_and_Fs.m.

That script writes as output the similarity matrices derived from the selected voxels, for each subject: haxby_wholebrain_selected_voxels_sims.mat

Those similarity matrices then serve as input for the across-subject decoding script:
Step4_AcrossSubjDecoding_using_selected_voxels_sims.m
That script is exactly the same as the one above from the first analysis,
Step2_AcrossSubjDecoding_SimMatrixPermutationMatching.m,
with the one alteration that a different .mat file of similarity matrix data is read-in at the beginning.

Because the .mat file to be read-in by this Step4 decoding script is included with the rest of the code, the reader may skip straight to running the Step4 script, if desired. The data in that .mat file can be recreated from scratch by running the Steps 1 to 3 scripts.

Chance-level performance of a Monte Carlo permutation distribution, compared against a binomial distribution

Chance performance for our new permutation-matching decoding approach is determined by a permutation distribution. For more standard multi-class decoding approaches, chance performance is given by a binomial distribution. The Matlab script perm_matching_vs_binomial.m in the Additional_scripts subdirectory calculates and compares these two distributions, for the case of eight different stimulus conditions. As running that code demonstrates, for both distributions the expected number correct for chance performance is 1 out of 8, and the $p<0.05$ critical-value number correct is 3 out of 8.

References

Hanke, M., Halchenko, Y. O., Sederberg, P. B., Olivetti, E., Fründ, I., Rieger, J. W., Herrmann, C. S., Haxby, J. V., Hanson, S. J., & Pollmann, S. (2009). PyMVPA: A unifying approach to the analysis of neuroscientific data. *Front Neuroinformatics*, *3*, 3.